Team Mini Project: Writing SQL Queries
DS450-01
Data Science Senior Capstone

Project Goal: To write simple to moderately complex SQL queries to extract data from a database using both a query interface (MySQL Workbench) and a Python script.

Scenario: You work for a small DVD rental company (mail order)[1]. The company has a database called Sakila that contains information about their films, rentals, customers, stores, etc. You have been asked to write a set of queries for the database so they can retrieve data from it.

Instructions:

We are going to go through the process of installing MySQL Server and MySQL Workbench on your computers in class. This will work best with Windows machines. Therefore, you might want to use a lab machine or if your partner has Windows use that machine. I will then do a quick refresher on SQL to get you started.

Part I

Write the SQL to complete the following queries. All SQL keywords should be capitalized. For example, SELECT * FROM customer;

SINGLE TABLE QUERIES

1. Get a list of all film titles alphabetized by title.
2. Find the description, release year, length, and rating for the movie "KENTUCKIAN GIANT".
3. Find the first name and last name of each employee (staff table). Your query should include the last name first, and then the first name.
4. Repeat the query above, but this time, the results should include only one column with the format `last name, first name`. The output column should be named "name"
5. Get the number of customers. The output should be a single number. Name the column "num_customers"
6. Get the number of customers who are active vs inactive in the system.
7. Get the average amount a customer spends on a rental.
8. Get maximum amount any customer has spent on a rental.
9. Get a list of the actors. The results should include only one column with the format `last name, first name.` The column should be named "actor_name" The results should be sorted be sorted alphabetically by the last name (ascending).
10. Repeat this query above, but the results should be in reverse order.
11. Repeat the query again, this time get only actors whose last names start with 'M' or 'V'. Order the results alphabetically by last name (ascending).
12. Repeat the query again, this time get only actors whose last names start with letters between 'M' and 'V' inclusive. Order the results alphabetically by last name (ascending).

[1] Yes, the still exist, although rare.

13. Get a list of each customer ID and the number of rentals they have in their history. Name this column Number of Rentals.

MULTI-TABLE QUERIES

1. Get a list of category names and a count of movies that fall into that category. Name the category column “category” the count column “num_films”. Order the results alphabetically (ascending). Use the WHERE clause to join the tables.
2. Repeat the query above using a JOIN clause instead of the WHERE clause.
3. Get a list of country names and a count of the cities that are in that country. Name the count column “num_cities”. Order the results alphabetically (ascending). Use the WHERE clause to join the tables.
4. Repeat the query above using a JOIN clause instead of the WHERE clause.
5. Get a list of each customer’s last name and first name and the number of rentals they have. Name the count column “num_rentals”. Order the result by the number of rentals in descending order. The highest number of rentals should be at the top. Sort any ties (same number of rentals) by last name (ascending). Use the WHERE clause to join the tables.
6. Repeat the query above using a JOIN clause instead of the WHERE clause.
7. Get a list of each customer’s last name and first name and the amount of money they have spent on rentals. Name the sum column “total_spent”. Order the result by the amount in descending order. The highest amount of money spent should be at the top. Sort any ties (amount of money spent) by last name (ascending). Use the JOIN clause for this query.
8. Get the number of actors in each film. Order the results (ascending) by the film title and name the column with the actor count “num_actors”.
9. Get the number of films each manager holds. Use only the manager staff id to identify the manager. Name the column with the number of films “num_films”.
10. Get the number of customers per manager. Use only the manager staff id to identify the manager. Name the column with the number of films “num_customers”. Order by store id (ascending).
11. Get the title and film category of each film. Order the results by category name. Rename the “name” column so it says “category”. This query will involve joining three tables using the JOIN syntax.
12. Get a list of each customer’s first and last name (individually, not concatenated) and their full address including city and country. Order the results by the customer’s last name. This will involve joining four tables using the JOIN syntax.
13. Repeat the query above except this time include only inactive customers from China.
14. Get a list of the titles of every film each customer has rented. Order the results by customer last name (ascending) and title (ascending).
15. Repeat the query above, but this time, include the category of each title in the results. Name the category column “category”. Order the results by the same columns (name and title).
16. Get a list of each customer that includes their first and last name, the number of rentals (num_rentals) they have had and the total amount (total_spent) of money they have spent on rentals. Order the results by last name (ascending).

17. Repeat the query above, but this time add the customer's country to the output. The order of the columns should be last_name, first_name, country, num_rentals, total_spent. Order rows by last name (ascending)

Part II

Now that you have written several moderately complicated queries, you have been asked to generate some plots/charts using Python/Pandas/Matplotlib/Seaborn to visualize some of the data and perform a linear regression. To do this, you will need to extract the data from the database using SQL and put it into a Pandas dataframe. I recommend you look at SQLAlchemy and the MySQL Python Connector to make the connection to the database running on your local machine. Then you can generate the required plots and analysis according to the specifications below. Ensure all plots have titles and axes are correctly labeled/styled.

1. Create a horizontal bar chart that shows all the film categories and how many films fall in that category. The bar chart should be sorted so the largest film category is at the top and the smallest is at the bottom.
2. Create a horizontal bar chart that shows the top ten countries by city count. The bar chart should be sorted so the country with the largest number of cities is at the top and the country with the smallest number of cities is at the bottom.
3. Create a horizontal bar chart that shows the top ten films by actor count. The bar chart should be sorted so the film with the largest number of actors is at the top and the film with the smallest number of actors is at the bottom.
4. Create a horizontal bar chart that shows the top ten countries by total number of rentals. The bar chart should be sorted so the country with the largest number of rentals is at the top and the country with the smallest number of rentals is at the bottom.
5. Create a horizontal bar chart that shows the top ten countries by total amount of money spent. The bar chart should be sorted so the country with the largest amount of money spent is at the top and the country with the smallest amount of money spent is at the bottom.
6. Create a scatter plot comparing the number of rentals on the X axis and the amount of money spent on the Y axis.

Finally, perform a statistical linear regression (not for prediction) with the number of rentals as the independent variable and the amount of money spent as the dependent variable. Report your results and your conclusions.

The output for this project should be:

- All SQL scripts/analysis files that you you used to perform your analysis. Ensure you document your SQL files so it is clear which queries match up with the questions above. Further ensure your notebook is well documented.
- A five to seven-minute video presentation of your project. In your presentation reflect on the tools, techniques and skills that were necessary to complete this project. What existing techniques and skills did you already have? What new skills did you need to obtain? What were the challenges for completing this project.

- All members of your team should be in the video. *Use a narrated PowerPoint to generate the video and upload it on video.bellarmine.edu. Include a link to the video in your README.md file under the heading* ***Sakila Database****.*

Submission: Create a new directory in your git repository of this project and put all your project files in that directory. Upload a link to your GitHub repository in the area provided on Moodle by the deadline specified.